Name: Krish Singh

Roll no: Bl.ai.u4aid25072

Lab Excercise 6

```
%qno 1
A=[-45 153 3;-18 60 1; 8 -26 3];
B=[-13 -42 98; 73 199 -469; 29 78 -184];
C=[-1 2 20 -22 2 2;
    -5 6 20 -22 2 2;
    -4 0 25 -22 2 2;
    -3 0 18 -16 2 2;
    -2 0 12 -12 3 2;
    -1 0 6 -6 0 4];
a=eig(A) % Gives Eigen Value of A
```

```
a = 3×1
    9.0000
    5.0000
    4.0000
```

```
b=eig(B) % Gives Eigen Value of B
```

```
b = 3×1
    3.0000
    1.0000
   -2.0000
```

```
c=eig(C) % Gives Eigen Value of C
```

```
c = 6×1
    4.0000
    5.0000
    6.0000
    1.0000
    3.0000
    2.0000
```

```
[V,D]=eig(A); % V stores EigenVectors and D stores Eigen Values in diagonal Matrix
[L,M]=eig(B); % L stores EigenVectors and M stores Eigen Values in diagonal Matrix
[N,O]=eig(C); % N stores EigenVectors and O stores Eigen Values in diagonal Matrix
AEvector1=V(:,1)
```

```
AEvector1 = 3×1
    0.9231
    0.3297
   -0.1978
```

```
AEvector2=V(:,2)
```

```
AEvector2 = 3×1
    0.9045
    0.3015
   -0.3015
```

```
Aevector3=V(:,3)
```

```
Aevector3 = 3×1
   -0.8018
   -0.2673
    0.5345
```

```
EigVectorsB1=L(:,1)
```

```
EigVectorsB1 = 3×1
    0.8083
   -0.5774
   -0.1155
```

```
EigVectorsB2=L(:,2)
```

```
EigVectorsB2 = 3×1
   -0.4082
   -0.8165
   -0.4082
```

```
EigVectorsB3=L(:,3)
```

```
EigVectorsB3 = 3×1
    0.0000
    0.9191
    0.3939
```

```
EigVectorsC1=N(:,1)
```

```
EigVectorsC1 = 6×1
   -0.6290
   -0.5241
   -0.4193
   -0.3145
   -0.2097
   -0.1048
```

```
EigVectorsC2=N(:,2)
```

```
EigVectorsC2 = 6×1
    0.5590
    0.5590
    0.4472
    0.3354
    0.2236
    0.1118
```

```
EigVectorsC3=N(:,3)
```

```
EigVectorsC3 = 6×1
   -0.5080
   -0.5080
   -0.5080
   -0.3810
   -0.2540
   -0.1270
```

```
EigValueA1=D(1,1)
```

```
EigValueA1 = 
9.0000
```

```
EigValueA2=D(2,2)
```

```
EigValueA2 =
5.0000
```

```
EigValueA3=D(3,3)
```

```
EigValueA3 =
4.0000
```

```
EigValueB1=M(1,1)
```

```
EigValueB1 =
3.0000
```

```
EigValueB2=M(2,2)
```

```
EigValueB2 =
1.0000
```

```
EigValueB3=M(3,3)
```

```
EigValueB3 =
-2.0000
```

```
EigValueC1=O(1,1)
```

```
EigValueC1 =
4.0000
```

```
EigValueC2=O(2,2)
```

```
EigValueC2 =
5.0000
```

```
EigValueC3=O(3,3)
```

```
EigValueC3 =
6.0000
```

```
EigValueC4=O(4,4)
```

```
EigValueC4 =
1.0000
```

```
EigValueC5=O(5,5)
```

```
EigValueC5 =
3.0000
```

```
EigValueC6=O(6,6)
```

```
EigValueC6 =
2.0000
```

```
%Qno 2 a
a1=trace(A);
b1=trace(D); % eigenvalues of A in diagonal matrix form
```

```
if (a1-b1<1e-6)
    disp("hence trace of matrix = sum of eigenvalues of the matrix");
else
    disp("Trace of Matrix A is not equal to sum of EigenValues of A");
end
```

```
hence trace of matrix = sum of eigenvalues of the matrix
```

```
a2=trace(B);
b2=trace(M); % eigenvalues of B in diagonal matrix form
if (a2-b2<1e-6)
    disp("hence trace of matrix = sum of eigenvalues of the matrix");
else
    disp("Trace of Matrix B is not equal to sum of EigenValues of B");
end
```

```
hence trace of matrix = sum of eigenvalues of the matrix
```

```
a3=trace(C);
b3=trace(O); % eigenvalues of C in diagonal matrix form
if (a3-b3<1e-6)
    disp("hence trace of matrix = sum of eigenvalues of the matrix");
else
    disp("Trace of Matrix C is not equal to sum of EigenValues of C");
end
```

```
hence trace of matrix = sum of eigenvalues of the matrix
```

```
%qno 2b
det1=det(A);
det2=det(B);
det3=det(C);
prodA=prod(diag(D));
prodB=prod(diag(M));
prodC=prod(diag(O));
if(det1-prodA<1e-6)
    disp("Determinant of A is equal to product of eigenvalues of A");
else
    disp("Determinant of A is not equal to product of eigenvalues of A");
end
```

```
Determinant of A is not equal to product of eigenvalues of A
```

```
if(det2-prodB<1e-6)
    disp("Determinant of B is equal to product of eigenvalues of B");
else
    disp("Determinant of B is not equal to product of eigenvalues of B");
end
```

```
Determinant of B is equal to product of eigenvalues of B
```

```
if(det3-prodC<1e-6)
```

```
    disp("Determinant of C is equal to product of eigenvalues of C");
else
    disp("Determinant of C is not equal to product of eigenvalues of C");
end
```

```
Determinant of C is equal to product of eigenvalues of C
```

```
%Qno 3
A=randi(10,7); %random int between 1-10
U=triu(A)
```

```
U = 7×7
     8     4     6     4     3     4     7
     0     3     9     9     5     1     5
     0     0     4     2     5     6    10
     0     0     0    10     8     4     2
     0     0     0     0     9     2     8
     0     0     0     0     0     3     8
     0     0     0     0     0     0     6
```

```
disp("EigenValues of A are :"); V=eig(U) %are eigen values in row vector form
```

```
EigenValues of A are :
V = 7×1
     8
     3
     4
    10
     9
     3
     6
```

```
for i=1:7
    fprintf('Eigenvalue %d =%f \n',i,V(i));
end
```

```
Eigenvalue 1 =8.000000
Eigenvalue 2 =3.000000
Eigenvalue 3 =4.000000
Eigenvalue 4 =10.000000
Eigenvalue 5 =9.000000
Eigenvalue 6 =3.000000
Eigenvalue 7 =6.000000
```

```
%Qno 4
V=randi(10,1,6) % generates a 1x6 matrix
```

```
V = 1×6
     8     1     9    10    10     9
```

```
A=fliplr(diag(V)) % makes the anti diagonal matrix with elements of V in anti
diagonal
```

```
A = 6×6
     0     0     0     0     0     8
     0     0     0     0     1     0
     0     0     0     9     0     0
```

```
     0     0    10     0     0     0
     0    10     0     0     0     0
     9     0     0     0     0     0
```

```
disp("Eigenevalues are = "); av=eig(A)
```

```
Eigenevalues are =
av = 6×1
    8.4853
   -8.4853
    9.4868
   -9.4868
    3.1623
   -3.1623
```

```
for i=1:6
    fprintf('Eigenvalue %d =%f \n',i,av(i));
end
```

```
Eigenvalue 1 =8.485281
Eigenvalue 2 =-8.485281
Eigenvalue 3 =9.486833
Eigenvalue 4 =-9.486833
Eigenvalue 5 =3.162278
Eigenvalue 6 =-3.162278
```

```
%Qno 5
B=randi(10,1,5)
```

```
B = 1×5
     9     9     3     3     6
```

```
C=fliplr(diag(B))
```

```
C = 5×5
     0     0     0     0     9
     0     0     0     9     0
     0     0     3     0     0
     0     3     0     0     0
     6     0     0     0     0
```

```
disp("Eigenvalues are :"); av=eig(C)
```

```
Eigenvalues are :
av = 5×1
    7.3485
   -7.3485
    5.1962
   -5.1962
    3.0000
```

```
for i=1:5
    fprintf('Eigenvalue %d =%f \n',i,av(i));
end
```

```
Eigenvalue 1 =7.348469
Eigenvalue 2 =-7.348469
Eigenvalue 3 =5.196152
```

```
Eigenvalue 4 =-5.196152
Eigenvalue 5 =3.000000
```

```
%qno 6
A=magic(5)
```

```
A = 5×5
    17    24     1     8    15
    23     5     7    14    16
     4     6    13    20    22
    10    12    19    21     3
    11    18    25     2     9
```

```
v=eig(A)
```

```
v = 5×1
   65.0000
  -21.2768
  -13.1263
   21.2768
   13.1263
```

```
disp(A)
```

```
    17    24     1     8    15
    23     5     7    14    16
     4     6    13    20    22
    10    12    19    21     3
    11    18    25     2     9
```

```
for i=1:5
    fprintf("Eigenvalue %d = %f \n",i,v(i));
end
```

```
Eigenvalue 1 = 65.000000
Eigenvalue 2 = -21.276765
Eigenvalue 3 = -13.126281
Eigenvalue 4 = 21.276765
Eigenvalue 5 = 13.126281
```

```
R1=A(1,:);
C1=A(:,1);

disp("When you generate a magic matrix of order 5 and find its eigenvalues, you
will notice that one eigenvalue is exactly equal to the common row and column sum")
```

```
When you generate a magic matrix of order 5 and find its eigenvalues, you will notice that one eigenvalue is exactl
```

```
disp("This shows that the vector of all ones is an eigenvector of the matrix and
that the magic matrix evenly distributes its weights across all rows and columns.")
```

```
This shows that the vector of all ones is an eigenvector of the matrix and that the magic matrix evenly distributes
```

```
%Qno 7 i,ii and iii for each matrices A,B and C
A=randi([1 10],4);
B=randi([1 10],10);
```

```
C=randi([1 10],15);
[Avector,Avalue]=eig(A);
[Bvector,Bvalue]=eig(B);
[Cvector,Cvalue]=eig(C);
%For matrix A
tra=trace(A)
```

```
tra =
26
```

```
SumA=sum(diag(Avalue))
```

```
SumA =
26.0000
```

```
if(tra-SumA<1e-6)
    disp("Trace of A is Equal to Sum of Eigenvalues of A");
else
    disp("Trace of A is not equal to Sum of eigenvalues of A");
end
```

```
Trace of A is Equal to Sum of Eigenvalues of A
```

```
detA=det(A)
```

```
detA =
291.0000
```

```
prodA=prod(diag(Avalue))
```

```
prodA =
2.9100e+02 + 1.4211e-14i
```

```
if(detA-prodA<1e-6)
    disp("Determinant of A is equal to Product of Eigenvalues of A")
else
    disp("Determinant of A is not equal to Product of Eigenvalues of A")
end
```

```
Determinant of A is equal to Product of Eigenvalues of A
```

```
eigA = diag(Avalue);
if all(isreal(eigA))
    disp("All eigenvalues of A are real");
else
    % To Check if complex eigenvalues come in conjugate pairs
    if norm(sort(eigA) - sort(conj(eigA))) < 1e-6
        disp("Complex eigenvalues of A appear in conjugate pairs");
    else
        disp("Complex eigenvalues of A do NOT appear properly in pairs");
    end
end
```

```
Complex eigenvalues of A appear in conjugate pairs
```

```
SumB=sum(diag(Bvalue))
```

```
SumB =
54.0000
```

```
trB=trace(B)
```

```
trB =
54
```

```
if(trB-SumB<1e-6)
    disp("Trace of B is equal to sum of EigenValues of B");
else
    disp("Trace of B is not equal to sum of eigenvalues of B");
end
```

```
Trace of B is equal to sum of EigenValues of B
```

```
DetB=det(B)
```

```
DetB =
-588610299
```

```
ProdB=prod(diag(Bvalue))
```

```
ProdB =
-5.8861e+08 + 7.5442e-08i
```

```
if abs(DetB - ProdB) < 1e-6 * max(1, abs(DetB))
    disp("Determinant of B is equual to product of eigenvalues");
else
    disp("Determinant of B is not equal to product of eigenvalues");
end
```

```
Determinant of B is equual to product of eigenvalues
```

```
eigB=diag(Bvalue);
if all(isreal(eigB))
    disp("All eigenvalues are real");
else
    if norm(sort(eigB)-sort(conj(eigB))) < 1e-6
        disp("Complex Eigenvalues of matrix B appear in pairs");
    else
        disp("complex Eigenvalues of matrix B doesnot appear in pairs");
    end
end
```

```
Complex Eigenvalues of matrix B appear in pairs
```

```
trC=trace(C)
```

```
trC =
85
```

```
SumC=sum(diag(Cvalue))
```

```
SumC =
85.0000
```

```
if(trC-SumC<1e-6)
    disp("Trace of matrix C is equal to Sum of EigenValues of C");
else
    disp("Trace of matrix C is not equal to Sum of Eigenvalues of C")
end
```

```
Trace of matrix C is equal to Sum of EigenValues of C
```

```
prodC=prod(diag(Cvalue))
```

```
prodC =
-1.2445e+13 + 1.9531e-03i
```

```
detC=det(C)
```

```
detC =
-1.2445e+13
```

```
if abs(prodC-detC)<1e-6*max(1,abs(detC))
    disp("Determinant of C is equal to product of eigenvalues of C");
else
    disp("Determinant of C is not equal to product of eigenvalues of C");
end
```

```
Determinant of C is equal to product of eigenvalues of C
```

```
eigC=diag(Cvalue);
if all(isreal(eigC))
    disp("All eigen vectors are real");
else
    if norm(sort(eigC)-sort(conj(eigC)))<1e-6
        disp("Complex Eigenvalues of matrix C appear in pairs");
    else
        disp("Complex Eigenvalues of matrix C doesnot appear in proper pairs");
    end
end
```

```
Complex Eigenvalues of matrix C appear in pairs
```

```
%qno 7b
%eigenvalues of A,B,C
eigA = eig(A);
eigB = eig(B);
eigC = eig(C);
% eigenvalue of 5A, B/3, and inv(C)
eig5A = eig(5*A);
eigBdiv3 = eig(B/3);
eigCinv = eig(inv(C));
% Theoretical expectations
expected5A = 5 * eigA;
```

```
expectedBdiv3 = (1/3) * eigB;
expectedCinv = 1 ./ eigC; % remember inverse is also reciprocal
% verification / comparing
t=1e-6;
if norm(sort(eig5A)-sort(expected5A)) <t
    disp('Eigenvalues of 5A = 5 x Eigenvalues of A');
else
    disp('Eigenvalues of 5A != 5 × Eigenvalues of A');
end
```

```
Eigenvalues of 5A = 5 x Eigenvalues of A
```

```
if norm(sort(eigBdiv3) - sort(expectedBdiv3)) < t
    disp('Eigenvalues of B/3 = (1/3) × Eigenvalues of B');
else
    disp('Eigenvalues of B/3 != (1/3) × Eigenvalues of B');
end
```

```
Eigenvalues of B/3 = (1/3) × Eigenvalues of B
```

```
if norm(sort(eigCinv) - sort(expectedCinv)) < t
    disp('Eigenvalues of inv(C) = 1 / Eigenvalues of C');
else
    disp('Eigenvalues of inv(C) != 1 / Eigenvalues of C');
end
```

```
Eigenvalues of inv(C) = 1 / Eigenvalues of C
```

```
%qno 7c
% Eigenvectors of A, B, C
[VA, ~] = eig(A);
[VB, ~] = eig(B);
[VC, ~] = eig(C);

% Eigenvectors of 5A, B/3, and inv(C)
[VA5, ~] = eig(5*A);
[VBdiv3, ~] = eig(B/3);
[VCinv, ~] = eig(inv(C));

% Verification / comparing
t = 1e-6;

% For A and 5A
if norm(abs(VA) - abs(VA5)) < t
    disp('Eigenvectors of 5A are same as those of A (direction unchanged)');
else
    disp('Eigenvectors of 5A differ from those of A');
end
```

```
Eigenvectors of 5A are same as those of A (direction unchanged)
```

```
% For B and B/3
if norm(abs(VB) - abs(VBdiv3)) < t
    disp('Eigenvectors of B/3 are same as those of B (direction unchanged)');
else
    disp('Eigenvectors of B/3 differ from those of B');
end
```

```
Eigenvectors of B/3 are same as those of B (direction unchanged)
```

```
% For C and inv(C)
if norm(abs(VC) - abs(VCinv)) < t
    disp('Eigenvectors of inv(C) are same as those of C (direction unchanged)');
else
    disp('Eigenvectors of inv(C) differ from those of C');
end
```

```
Eigenvectors of inv(C) differ from those of C
```

```
disp('Scalar multiplication or inversion of a matrix does not change its
eigenvector directions.');
```

```
Scalar multiplication or inversion of a matrix does not change its eigenvector directions.
```

```
disp('Only the eigenvalues get scaled or reciprocated accordingly.');
```

```
Only the eigenvalues get scaled or reciprocated accordingly.
```

```
%Qno 7d
S=A+transpose(A)
```

```
S = 4×4
    12     9    12    14
     9    16    10    12
    12    10    16     9
    14    12     9     8
```

```
[VE,VA]=eig(S)
```

```
VE = 4×4
    0.5838   -0.5759    0.2638    0.5079
    0.2438    0.3554   -0.7447    0.5096
   -0.1214    0.6032    0.5989    0.5126
   -0.7649   -0.4220   -0.1310    0.4687
VA = 4×4
   -5.0808         0         0         0
         0    4.1372         0         0
         0         0    6.8816         0
         0         0         0   46.0621
```

```
for i=1:4
    fprintf("Eigenvector %d is : ",i);
    disp(VE(:,i));
```

```
end
```

```
Eigenvector 1 is :
    0.5838
    0.2438
   -0.1214
   -0.7649
Eigenvector 2 is :
   -0.5759
    0.3554
    0.6032
   -0.4220
Eigenvector 3 is :
    0.2638
   -0.7447
    0.5989
   -0.1310
Eigenvector 4 is :
    0.5079
    0.5096
    0.5126
    0.4687
```

```
for i=1:4
    fprintf("Eigenvalue %d is : %f \n",i,VA(i,i))
end
```

```
Eigenvalue 1 is : -5.080815
Eigenvalue 2 is : 4.137193
Eigenvalue 3 is : 6.881566
Eigenvalue 4 is : 46.062056
```

```
%Qno 8
mat=randi([1,10],3)
```

```
mat = 3×3
     9    10     3
    10     7     6
     2     1    10
```

```
[R,D,L] = eig(mat)
```

```
R = 3×3
    0.6823   -0.7062   -0.5253
   -0.7291   -0.6728   -0.2182
   -0.0533   -0.2204    0.8225
D = 3×3
   -1.9198         0         0
         0   19.4625         0
         0         0    8.4573
L = 3×3
    0.6387   -0.6302   -0.1248
   -0.7398   -0.5497   -0.1881
    0.2116   -0.5483    0.9742
```

```
for i=1:3
    fprintf("Right EigenVector %d is :",i);
    disp(R(:,i));
end
```

```
Right EigenVector 1 is :
    0.6823
   -0.7291
   -0.0533
Right EigenVector 2 is :
   -0.7062
   -0.6728
   -0.2204
Right EigenVector 3 is :
   -0.5253
   -0.2182
    0.8225
```

```
for i=1:3
    fprintf("Left EigenVector %d is :",i);
    disp(L(:,i));
end
```

```
Left EigenVector 1 is :
    0.6387
   -0.7398
    0.2116
Left EigenVector 2 is :
   -0.6302
   -0.5497
   -0.5483
Left EigenVector 3 is :
   -0.1248
   -0.1881
    0.9742
```

```
for i=1:3
    fprintf("Eigenvalue %d is : %f \n",i,D(i,i));
end
```

```
Eigenvalue 1 is : -1.919800
Eigenvalue 2 is : 19.462479
Eigenvalue 3 is : 8.457322
```

```
%qno 8b
syms x;
p=charpoly(mat,x)
```

p = $x^3 - 26\,x^2 + 111\,x + 316$

```
%qno 8c
PA=polyvalm(sym2poly(p),mat)
```

```
PA = 3×3
     0     0     0
     0     0     0
     0     0     0
```

```
if norm(PA,'fro')<1e-6
    disp("Value obtained is 0 ; hence satisfying Cayley-Hamilton theorem");
else
    disp("Value is not Zero ; doesnot satisfy Cayley-Hamilton theorem");
end
```

```
Value obtained is 0 ; hence satisfying Cayley-Hamilton theorem
```

```
%trying it with other square matrix
A=randi([1,10],4);
syms x;
c=charpoly(A,x)
```

c = $x^4 - 20\,x^3 - 69\,x^2 + 76\,x - 276$

```
PA=polyvalm(sym2poly(c),A)
```

```
PA = 4×4
     0     0     0     0
     0     0     0     0
     0     0     0     0
     0     0     0     0
```

```
if norm(PA,'fro')<1e-6
    disp("Value obtained is 0 ; hence satisfying Cayley-Hamilton theorem");
else
    disp("Value is not Zero ; doesnot satisfy Cayley-Hamilton theorem");
end
```

```
Value obtained is 0 ; hence satisfying Cayley-Hamilton theorem
```

```
%qno 9
A=randi([1,10],5,3);
c=A*A'
```

```
c = 5×5
    65    25   101    79    67
    25    51    58    80    58
   101    58   166   148   121
    79    80   148   182   143
    67    58   121   143   114
```

```
d=A'*A
```

```
d = 3×3
    77   111    76
   111   263   219
    76   219   238
```

```
EignC=eig(c)
```

```
EignC = 5×1
   -0.0000
    0.0000
   13.4541
   53.8863
  510.6596
```

```
EignD=eig(d)
```

```
EignD = 3×1
```

```
   13.4541
   53.8863
  510.6596
```

```
disp("Non zero Eigenvalues of AA' and A'A is same");
```

```
Non zero Eigenvalues of AA' and A'A is same
```

```
%qno 10
ev1=3;ev2=5;ev3=7;ev4=11;
evec1=[1;1;2;2];
evec2=[1;0;1;0];
evec3=[2;1;2;0];
evec4=[0;1;1;0];
V=[evec1,evec2,evec3,evec4];
D=diag([ev1,ev2,ev3,ev4]);
disp("The formed matrix is : "); A=V*D*inv(V)
```

```
The formed matrix is :
A = 4×4
     9     4    -4    -1
    -4     7     4    -4
    -2     4     7    -5
     0     0     0     3
```

```
%Qno 11
A=randi([1,10],5,2)*randi([1,10],2,5)
```

```
A = 5×5
    95   143   146    79   132
    56    84    86    46    78
   105   161   162    93   144
    86   138   134    88   114
    64    90    97    41    93
```

```
rA=rank(A)
```

```
rA =
2
```

```
%Qno 11 a
B=A+A'
```

```
B = 5×5
   190   199   251   165   196
   199   168   247   184   168
   251   247   324   227   241
   165   184   227   176   155
   196   168   241   155   186
```

```
rB=rank(B)
```

```
rB =
4
```

```
%Qno 11 b
[V,D]=eig(B)
```

```
V = 5×5
    0.5370    0.3880   -0.5342    0.2956    0.4341
   -0.7422    0.1528   -0.4329   -0.2504    0.4193
    0.1341   -0.8177    0.0089    0.0245    0.5592
    0.2771    0.2973    0.4207   -0.7097    0.3926
   -0.2571    0.2630    0.5918    0.5879    0.4110
D = 5×5
```

10^3 ×

```
   -0.0311         0         0         0         0
         0   -0.0013         0         0         0
         0         0    0.0000         0         0
         0         0         0    0.0359         0
         0         0         0         0    1.0405
```

```
for i=1:5
    fprintf("Eigen vector %d is :",i);
    disp(V(:,i));
end
```

```
Eigen vector 1 is :
    0.5370
   -0.7422
    0.1341
    0.2771
   -0.2571
Eigen vector 2 is :
    0.3880
    0.1528
   -0.8177
    0.2973
    0.2630
Eigen vector 3 is :
   -0.5342
   -0.4329
    0.0089
    0.4207
    0.5918
Eigen vector 4 is :
    0.2956
   -0.2504
    0.0245
   -0.7097
    0.5879
Eigen vector 5 is :
    0.4341
    0.4193
    0.5592
    0.3926
    0.4110
```

```
for i=1:5
    fprintf("Eigennvalue %d is : %f \n",i,D(i,i));
end
```

```
Eigennvalue 1 is : -31.089362
Eigennvalue 2 is : -1.308898
Eigennvalue 3 is : 0.000000
Eigennvalue 4 is : 35.943652
Eigennvalue 5 is : 1040.454608
```

```
%Qno 11 c
```

```
if all(isreal(D))
    disp("All Eigen Values are Real")
else
    disp("There are complex Eigen Values")
end
```

```
All Eigen Values are Real
```

```
fprintf('A symmetric matrix has real eigenvalues because : \n transpose(x)*A*x is
always real for any real vector x, and by eigenvalue equation Ax=Lamdba.x \n this
forces lambda to be real')
```

```
A symmetric matrix has real eigenvalues because :
 transpose(x)*A*x is always real for any real vector x, and by eigenvalue equation Ax=Lamdba.x
 this forces lambda to be real
```

```
%Qno 11 d
Di=diag(D);
cnt=0;
for i=1:5
    if abs(Di(i))<1e-6||abs(Di(i))<-1e-6
        cnt=cnt+1;
    else
        continue;
    end
end
fprintf("Number of Zero Eigenvalues is %d",cnt);
```

```
Number of Zero Eigenvalues is 1
```

```
%Qno 11 e
check=V'*V
```

```
check = 5×5
    1.0000    0.0000    0.0000    0.0000    0.0000
    0.0000    1.0000   -0.0000   -0.0000   -0.0000
    0.0000   -0.0000    1.0000   -0.0000   -0.0000
    0.0000   -0.0000   -0.0000    1.0000   -0.0000
    0.0000   -0.0000   -0.0000   -0.0000    1.0000
```

```
if norm(check-eye(size(check)))<1e-6
    disp("Eigenvectors of B are orthogonal");
else
    disp("Eigenvectors of b are not orthogonal");
end
```

```
Eigenvectors of B are orthogonal
```

```
%Qno 12
A=randi([1,10],10,3)*randi([1,10],3,5)
```

```
A = 10×5
    52   151   144    81    50
    79   166   158    97    53
    27    81    58    44    57
```

```
    26   129   112    63    62
    20    91    80    45    42
    27    97    86    50    43
    42   102    80    58    60
    51   175   156    91    76
    31   142   106    71    95
    61   143   114    82    81
```

```
rA=rank(A)
```

```
rA =
3
```

```
%Qno 12 a
S1=A*A'
```

```
S1 = 10×10
       55302       62433       28401       45162       32046       34635 ···
       62433       70979       32032       50561       35917       38952
       28401       32032       15839       23953       16925       18225
       45162       50561       23953       37674       26658       28663
       32046       35917       16925       26658       18870       20303
       34635       38952       18225       28663       20303       21883
       36804       41696       20008       30584       21652       23388
       62712       70582       32936       51818       36712       39586
       48819       54691       27026       41359       29207       31362
       51873       58816       28067       42989       30445       32905
```

```
S2=A'*A
```

```
S2 = 5×5
       20526       57280       49926       31231       26221
       57280      172891      149554       92618       81449
       49926      149554      130372       80208       68842
       31231       92618       80208       49810       43428
       26221       81449       68842       43428       40977
```

```
%Qno 12 b
rs1=rank(S1)
```

```
rs1 =
3
```

```
rs2=rank(S2)
```

```
rs2 =
3
```

```
%Qno 12 c
es1=eig(S1)
```

es1 = 10×1

10^5 ×

```
  -0.0000
  -0.0000
  -0.0000
  -0.0000
   0.0000
   0.0000
   0.0000
   0.0131
```

```
    0.0388
    4.0939
```

```
es2=eig(S2)
```

```
es2 = 5×1
10^5 ×
   -0.0000
    0.0000
    0.0131
    0.0388
    4.0939
```

```
fprintf("What i noticed is non zero eigen values of A*A' and A'*A are Equal and
their count is equal to the rank of A")
```

```
What i noticed is non zero eigen values of A*A' and A'*A are Equal and their count is equal to the rank of A
```

```
%Qno 13
A=randi([1,10],7,2)*randi([1,10],2,7)
```

```
A = 7×7
    23    13    27    16    15    26    33
   125    88   123    55   108   113   120
   106    69   112    57    83   105   121
    92    71    82    29    89    73    67
    85    72    67    15    92    57    40
    96    65    98    47    79    91   101
    85    72    67    15    92    57    40
```

```
rank(A)
```

```
ans =
2
```

```
%Qno 13 a
B=A-A'
```

```
B = 7×7
     0  -112   -79   -76   -70   -70   -52
   112     0    54   -16    36    48    48
    79   -54     0   -25    16     7    54
    76    16    25     0    74    26    52
    70   -36   -16   -74     0   -22   -52
    70   -48    -7   -26    22     0    44
    52   -48   -54   -52    52   -44     0
```

```
rank(B)
```

```
ans =
4
```

```
%Qno 13 b
[V,D]=eig(B)
```

```
V = 7×7 complex
   0.5601 + 0.0317i   0.5601 - 0.0317i   0.1595 - 0.1595i   0.1595 + 0.1595i ···
  -0.1614 - 0.3844i  -0.1614 + 0.3844i  -0.1010 - 0.3357i  -0.1010 + 0.3357i
   0.0587 - 0.3099i   0.0587 + 0.3099i   0.3312 - 0.1125i   0.3312 + 0.1125i
```

```
  -0.1819 - 0.3084i  -0.1819 + 0.3084i  -0.0521 + 0.2281i  -0.0521 - 0.2281i
   0.2555 - 0.1826i   0.2555 + 0.1826i  -0.5321 - 0.1481i  -0.5321 + 0.1481i
   0.0710 - 0.2787i   0.0710 + 0.2787i   0.2523 - 0.0025i   0.2523 + 0.0025i
   0.2312 - 0.2215i   0.2312 + 0.2215i  -0.0715 + 0.5246i  -0.0715 - 0.5246i
D = 7×7 complex
10^2 ×
   0.0000 + 2.4066i   0.0000 + 0.0000i   0.0000 + 0.0000i   0.0000 + 0.0000i ...
   0.0000 + 0.0000i   0.0000 - 2.4066i   0.0000 + 0.0000i   0.0000 + 0.0000i
   0.0000 + 0.0000i   0.0000 + 0.0000i   0.0000 + 0.7778i   0.0000 + 0.0000i
   0.0000 + 0.0000i   0.0000 + 0.0000i   0.0000 + 0.0000i   0.0000 - 0.7778i
   0.0000 + 0.0000i   0.0000 + 0.0000i   0.0000 + 0.0000i   0.0000 + 0.0000i
   0.0000 + 0.0000i   0.0000 + 0.0000i   0.0000 + 0.0000i   0.0000 + 0.0000i
   0.0000 + 0.0000i   0.0000 + 0.0000i   0.0000 + 0.0000i   0.0000 + 0.0000i
```

```matlab
%Qno 13 c
for i=1:7
    fprintf("Eigen vector %d is :",i);
    disp(V(:,1));
end
```

```
Eigen vector 1 is :
   0.5601 + 0.0317i
  -0.1614 - 0.3844i
   0.0587 - 0.3099i
  -0.1819 - 0.3084i
   0.2555 - 0.1826i
   0.0710 - 0.2787i
   0.2312 - 0.2215i
Eigen vector 2 is :
   0.5601 + 0.0317i
  -0.1614 - 0.3844i
   0.0587 - 0.3099i
  -0.1819 - 0.3084i
   0.2555 - 0.1826i
   0.0710 - 0.2787i
   0.2312 - 0.2215i
Eigen vector 3 is :
   0.5601 + 0.0317i
  -0.1614 - 0.3844i
   0.0587 - 0.3099i
  -0.1819 - 0.3084i
   0.2555 - 0.1826i
   0.0710 - 0.2787i
   0.2312 - 0.2215i
Eigen vector 4 is :
   0.5601 + 0.0317i
  -0.1614 - 0.3844i
   0.0587 - 0.3099i
  -0.1819 - 0.3084i
   0.2555 - 0.1826i
   0.0710 - 0.2787i
   0.2312 - 0.2215i
Eigen vector 5 is :
   0.5601 + 0.0317i
  -0.1614 - 0.3844i
   0.0587 - 0.3099i
  -0.1819 - 0.3084i
   0.2555 - 0.1826i
   0.0710 - 0.2787i
   0.2312 - 0.2215i
Eigen vector 6 is :
```

```
   0.5601 + 0.0317i
  -0.1614 - 0.3844i
   0.0587 - 0.3099i
  -0.1819 - 0.3084i
   0.2555 - 0.1826i
   0.0710 - 0.2787i
   0.2312 - 0.2215i
Eigen vector 7 is :
   0.5601 + 0.0317i
  -0.1614 - 0.3844i
   0.0587 - 0.3099i
  -0.1819 - 0.3084i
   0.2555 - 0.1826i
   0.0710 - 0.2787i
   0.2312 - 0.2215i
```

```
for i=1:7
    fprintf("Eigenvalue %d is:",i)
    disp(D(i,i));
end
```

```
Eigenvalue 1 is:
   0.0000e+00 + 2.4066e+02i
Eigenvalue 2 is:
   0.0000e+00 - 2.4066e+02i
Eigenvalue 3 is:
   0.0000 +77.7839i
Eigenvalue 4 is:
   0.0000 -77.7839i
Eigenvalue 5 is:
   0.0000e+00 + 2.9332e-16i
Eigenvalue 6 is:
   0.0000e+00 - 2.9332e-16i
Eigenvalue 7 is:
     0
```

```
EigD=diag(D);
if all(isreal(EigD))
    disp("There are real eigenvalues")
else
    disp("EigenValues are imaginary or zero for skew symmetric matrices there")
end
```

```
EigenValues are imaginary or zero for skew symmetric matrices there
```

```
fprintf("A real skew-symmetric matrix always has purely imaginary eigenvalues and
possibly zero. \n This happens because the matrix is equal to the negative of its
transpose which forces the eigenvalues to lie on the imaginary axis in the complex
plane.")
```

```
A real skew-symmetric matrix always has purely imaginary eigenvalues and possibly zero.
 This happens because the matrix is equal to the negative of its transpose which forces the eigenvalues to lie on th
```

```
%Qno 13 d
cnt=0
```

```
cnt =
0
```

```
for i=1:7
    if(EigD(i)==0)
        cnt=cnt+1;
    else
        continue
    end
end
fprintf("number of zero eigenvalues is : %d",cnt)
```

```
number of zero eigenvalues is : 1
```

```
fprintf("Every odd-order skew-symmetric matrix (like 3×3, 5×5, 7×7 etc) always
has at least one zero eigenvalue.\nThis happens because its determinant is always
zero.")
```

```
Every odd-order skew-symmetric matrix (like 3×3, 5×5, 7×7 etc) always has at least one zero eigenvalue.
This happens because its determinant is always zero.
```

```
%Qno 13 e
A=randi([1,10],8)
```

```
A = 8×8
     7     2     8     8    10     6     6     6
     3     2     4     4     2     3     6     6
     6     2     5     6     3     5     2     3
     9     1     3     5     5    10    10     1
     6     1     8     1     3     3     4     9
     8     6     2     9     2     1     7     1
     9     7     8     7     8    10     4     4
     9    10     9     3     2     7     1     8
```

```
C=A'-A
```

```
C = 8×8
     0     1    -2     1    -4     2     3     3
    -1     0    -2    -3    -1     3     1     4
     2     2     0    -3     5    -3     6     6
    -1     3     3     0    -4    -1    -3     2
     4     1    -5     4     0    -1     4    -7
    -2    -3     3     1     1     0     3     6
    -3    -1    -6     3    -4    -3     0    -3
    -3    -4    -6    -2     7    -6     3     0
```

```
D=eig(C)
```

```
D = 8×1 complex
   0.0000 +15.3642i
   0.0000 -15.3642i
   0.0000 + 9.0178i
   0.0000 - 9.0178i
   0.0000 + 5.6231i
   0.0000 - 5.6231i
   0.0000 + 0.0295i
   0.0000 - 0.0295i
```

```
for i=1:8
    fprintf("eigenvalue %d is :",i)
    disp(D(i))
```

```
end
```

```
eigenvalue 1 is :
   0.0000 +15.3642i
eigenvalue 2 is :
   0.0000 -15.3642i
eigenvalue 3 is :
   0.0000 + 9.0178i
eigenvalue 4 is :
   0.0000 - 9.0178i
eigenvalue 5 is :
   0.0000 + 5.6231i
eigenvalue 6 is :
   0.0000 - 5.6231i
eigenvalue 7 is :
   0.0000 + 0.0295i
eigenvalue 8 is :
   0.0000 - 0.0295i
```

```
cnt=0;
for i=1:8
    if(D(i)==0)
        cnt=cnt+1;
    end
end
fprintf("Number of zero eigenvalues is : %d",cnt)
```

```
Number of zero eigenvalues is : 0
```

```
fprintf("Here number of zero eigenvalues for 8x8 skew symmetric matrix = 0 because:
\nFor even-order skew-symmetric matrices, the determinant may not be zero, so they
may or may not have a zero eigenvalue.")
```

```
Here number of zero eigenvalues for 8x8 skew symmetric matrix = 0 because:
For even-order skew-symmetric matrices, the determinant may not be zero, so they may or may not have a zero eigenva
```

```
%qno 14
V=randi([1,10],1,10);
disp(V)
```

```
     3     8     6     4     8     2     1     6     7     6
```

```
%Qno 14 a
Z=null(V)
```

```
Z = 10×9
   -0.4507   -0.3381   -0.2254   -0.4507   -0.1127   -0.0563   -0.3381   -0.3944 ···
    0.8262   -0.1303   -0.0869   -0.1738   -0.0434   -0.0217   -0.1303   -0.1521
   -0.1303    0.9022   -0.0652   -0.1303   -0.0326   -0.0163   -0.0978   -0.1141
   -0.0869   -0.0652    0.9566   -0.0869   -0.0217   -0.0109   -0.0652   -0.0760
   -0.1738   -0.1303   -0.0869    0.8262   -0.0434   -0.0217   -0.1303   -0.1521
   -0.0434   -0.0326   -0.0217   -0.0434    0.9891   -0.0054   -0.0326   -0.0380
   -0.0217   -0.0163   -0.0109   -0.0217   -0.0054    0.9973   -0.0163   -0.0190
   -0.1303   -0.0978   -0.0652   -0.1303   -0.0326   -0.0163    0.9022   -0.1141
   -0.1521   -0.1141   -0.0760   -0.1521   -0.0380   -0.0190   -0.1141    0.8669
   -0.1303   -0.0978   -0.0652   -0.1303   -0.0326   -0.0163   -0.0978   -0.1141
```

```
sizen=size(Z)
```

```
sizen = 1×2
    10     9
```

```
fprintf("Number of independent orthogonal vectors = %d",sizen(2))
```

```
Number of independent orthogonal vectors = 9
```

```
%Qno 14 b
W=V'*V
```

```
W = 10×10
     9    24    18    12    24     6     3    18    21    18
    24    64    48    32    64    16     8    48    56    48
    18    48    36    24    48    12     6    36    42    36
    12    32    24    16    32     8     4    24    28    24
    24    64    48    32    64    16     8    48    56    48
     6    16    12     8    16     4     2    12    14    12
     3     8     6     4     8     2     1     6     7     6
    18    48    36    24    48    12     6    36    42    36
    21    56    42    28    56    14     7    42    49    42
    18    48    36    24    48    12     6    36    42    36
```

```
%Qno 14 c
rw=rank(W)
```

```
rw =
1
```

```
evwn=eig(W)
```

```
evwn = 10×1
   -0.0000
   -0.0000
   -0.0000
   -0.0000
    0.0000
    0.0000
    0.0000
    0.0000
    0.0000
  315.0000
```

```
en=eig(W)
```

```
en = 10×1
   -0.0000
   -0.0000
   -0.0000
   -0.0000
    0.0000
    0.0000
    0.0000
    0.0000
    0.0000
  315.0000
```

```
%Qno 14 d
U=V*V'
```

```
U =
315
```

```
%Qno 14 e
t1=trace(U)
```

```
t1 =
315
```

```
t2=trace(W)
```

```
t2 =
315
```

```
nv=norm(V)^2
```

```
nv =
315.0000
```

```
if(t1==t2)&&(t2-nv<1e-6)
    disp("Hence we prove : trace(V*V')=trace(V'*V)=||V||^2")
else
    disp("Verification failed")
end
```

```
Hence we prove : trace(V*V')=trace(V'*V)=||V||^2
```

```
%Qno 15
v1=[1;1];
v2=[1;-1];
u1=v1/norm(v1);
u2=v2/norm(v2);
P = [u1,u2];
D = diag([5,9]);

M = P*D*P'   % or simply M = [7 -2; -2 7]
```

```
M = 2×2
    7.0000   -2.0000
   -2.0000    7.0000
```

```
fprintf(" To verify : \n M*u1 should be = 5*u1")
```

```
 To verify :
 M*u1 should be = 5*u1
```

```
disp(M*u1)
```

```
    3.5355
    3.5355
```

```
disp(5*u1)
```

```
    3.5355
    3.5355
```

```
fprintf(" To verify : \n M*u2 should be = 9*u2")
```

```
To verify :
M*u2 should be = 9*u2
```

```
disp(M*u2)
```

```
    6.3640
   -6.3640
```

```
disp(9*u2)
```

```
    6.3640
   -6.3640
```

```
%Cross check for Errors
[V,D]=eig(M)
```

```
V = 2×2
   -0.7071   -0.7071
   -0.7071    0.7071
D = 2×2
    5.0000         0
         0    9.0000
```

```
fprintf("Eigenvalues should be 5 and 9 : \n"), disp(diag(D));
```

```
Eigenvalues should be 5 and 9 :
    5.0000
    9.0000
```